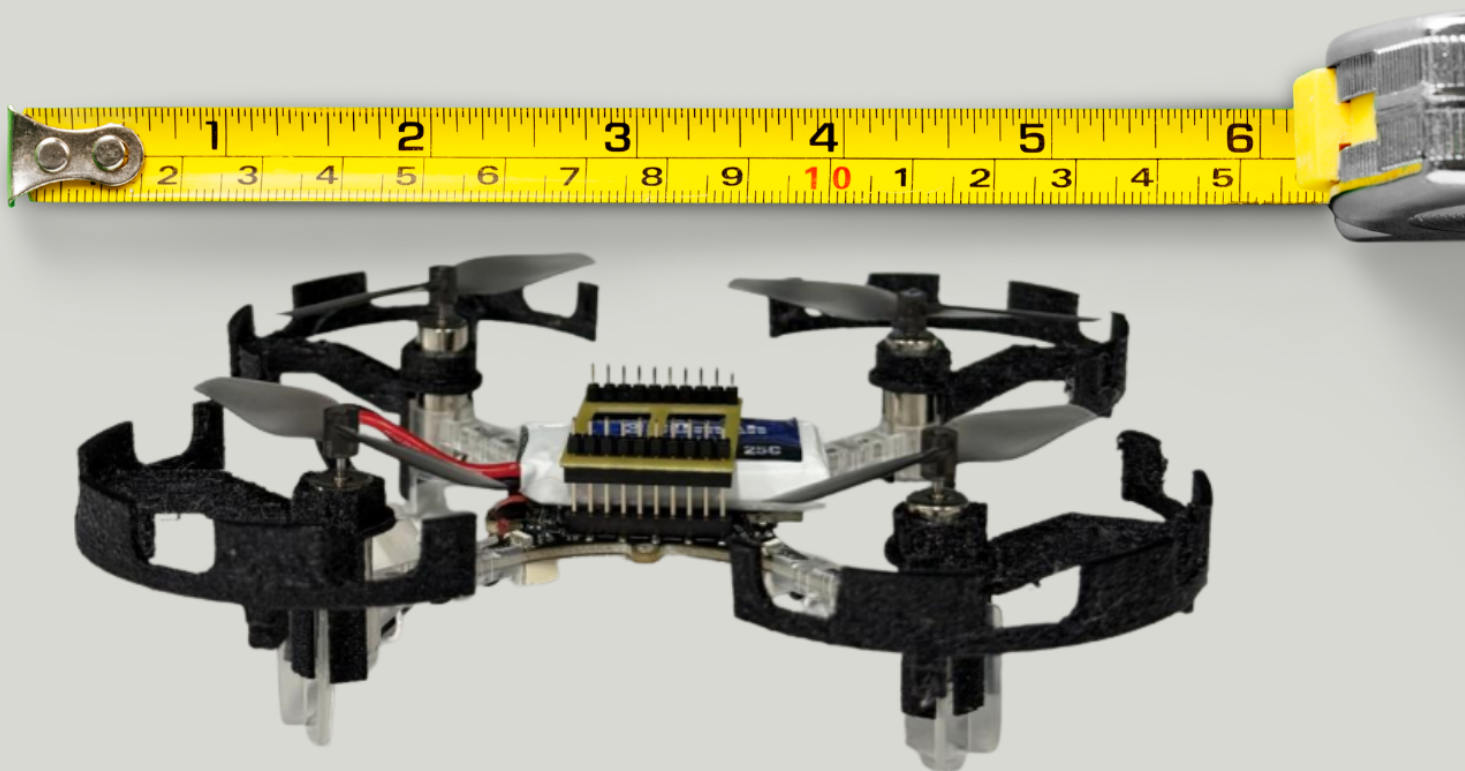

Desarrollo de infraestructura experimental para la mejora en eficiencia y seguridad de drones individuales CrazyFlie mediante Python/Matlab

Angela María Figueroa Pérez



UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Desarrollo de infraestructura experimental para la mejora en
eficiencia y seguridad de drones individuales CrazyFlie
mediante Python/Matlab**

Trabajo de graduación presentado por Angela María Figueroa Pérez
para optar al grado académico de Licenciada en Ingeniería Mecatrónica

Guatemala,

2025

Vo.Bo.:

(f) 

M.Sc. Miguel Enrique Zea Arenales

(f) 

M.Sc. Carlos Alberto Esquit Hernández

Fecha de aprobación: Guatemala, 28 de noviembre de 2025.

Este trabajo de graduación se lo dedico con todo mi cariño a mi mamá y a mi papá, quienes siempre creyeron en mí. Su esfuerzo, apoyo y amor incondicional han sido fundamentales en mi formación personal y profesional. Este logro también es de ustedes.

Asimismo, este trabajo fue posible gracias al apoyo de muchas personas e instituciones que me acompañaron a lo largo de este proceso. Agradezco de manera especial a la Universidad del Valle de Guatemala, por brindarme el espacio, los recursos y las herramientas necesarias para desarrollar esta investigación. También, expreso mi sincero agradecimiento a mis asesores y docentes, quienes con su guía, paciencia y orientación académica contribuyeron de manera fundamental a la realización de este proyecto.

Deseo reconocer el apoyo de mis familiares y amigos, quienes con su compañía, motivación y confianza me impulsaron a continuar, incluso en los momentos más desafiantes.

Prefacio	I
Índice de figuras	V
Resumen	VI
Abstract	VII
1. Introducción	1
2. Antecedentes	3
3. Justificación	6
4. Objetivos	7
4.1. Objetivo general	7
4.2. Objetivos específicos	7
5. Hipótesis	8
5.1. Declaración del problema	8
5.2. Relación entre variables	8
5.3. Comprobación y cuantificación	9
5.4. Delimitación del estudio	9
6. Marco teórico	10
6.1. CrazyFlie 2.1	10
6.2. Cargador inalámbrico Tozo W1	15
6.3. Motion capture marker deck	15
7. Diseño de guardas para hélices del dron Crazyflie 2.1	17
7.1. Enfoque general de diseño	17

8. Diseño e implementación de estaciones de carga para drones CrazyFlie	
2.1	25
8.1. Requerimientos funcionales del diseño	25
8.2. Diseño de los protectores de los cargadores inalámbricos Tozo	26
8.3. Proceso de modelado de la caja	26
8.4. Calibración del cargador Qi Tozo como marker en Robotat	32
8.5. Requerimientos para la generación del patrón de marker en la estación de carga	32
9. Optimización de herramientas de software para la comunicación y control del dron CrazyFlie 2.1	39
9.1. Integración de Matlab con el ecosistema Robotat	39
9.2. Procesamiento y transmisión eficiente de datos al dron CrazyFlie 2.1	40
9.3. Pruebas de llegada autónoma a la estación de carga	41
9.4. Evaluación del rendimiento y tiempo de respuesta del sistema	41
9.5. Comprobación de la carga en la estación de carga	42
10. Conclusiones	44
11. Recomendaciones	45
12. Referencias	46
13. Anexos	48
13.1. Código fuente	48

Índice de figuras

1.	Plataforma de pruebas para drones Crazyflie 2.0	3
2.	Vista frontal del ecosistema Robotat	4
3.	Pruebas de laboratorio con el dron CrazyFlie 2.1	5
4.	Dron CrazyFlie 2.1	10
5.	Sistema de ejes de un cuadricóptero	11
6.	Convenciones de rotación para los ángulos de actitud	12
7.	Comunicación inalámbrica del dron CrazyFlie 2.1	12
8.	Placa de expansión Flow Deck	13
9.	Qi 1.2 charger deck	14
10.	Tozo cargador inalámbrico actualizado 2025	15
11.	<i>Motion capture marker deck</i>	16
12.	<i>Passive marker deck</i> reflectantes utilizados en sistemas de captura de movimiento	16
13.	Motor sin núcleo de CC	18
14.	Paquete de motor de CC (mejora de empuje)	18
15.	Soportes del motor	18
16.	Hélice 47-17 (4CCW+4CW) color negro	19
17.	HQ Ultralight Prop 2X0.9 (51MM) Grey(2CW+2CCW)-PC-1MM Shaft	19
18.	Dimensiones del dron CrazyFlie 2.1	20
19.	Primer diseño	21
20.	Segundo diseño	21
21.	Tercer diseño	22
22.	Dron CrazyFlie 2.1 con guardas instaladas	23
23.	Diez drones CrazyFlie 2.1 con guardas instaladas	23
24.	Diseño de la guarda de protección para las hélices	24
25.	Dron con guarda integrada	24
26.	Estuche para cargadores Tozo	27
27.	Estación de carga para cargadores Tozo (versión 2)	27
28.	Estación de carga física para cargadores Tozo (versión 2)	28
29.	Representación dimensional de la caja de la estación de carga.	28

30.	Base de estación de carga	29
31.	Llave para asegurar la caja de estación de carga	30
32.	Estación de carga para cargadores Tozo (versión 3)	30
33.	Caja completa de la estación de carga final	31
34.	Estación de carga final con patrón	32
35.	Cinta adhesiva reflectiva	33
36.	Plantilla de círculos	33
37.	Patrón para estación de carga	34
38.	Patrones para estación de carga	35
39.	Patrón en estación de carga	36
40.	Marker 90 visualizado en Robotat	36
41.	Trayectoria del marker 27 y conexión con marker 90	37
42.	Trayectoria del dron CrazyFlie 2.1 con el sistema MQTT	42
43.	Comprobación visual de carga en la estación del Crazyflie 2.1	43
44.	Instalación del Qi 1.2 charger deck en el dron Crazyflie 2.1	43

El presente trabajo documenta el desarrollo e integración de una infraestructura experimental destinada a mejorar la seguridad del dron CrazyFlie 2.1 dentro del entorno del ecosistema Robotat de la Universidad del Valle de Guatemala. La investigación se desarrolló en tres articulaciones principales: el diseño y optimización de las guardas protectoras, cuya masa llegó a ser de 7 g, lo que favoreció la capacidad del dron de despegar y redujo daños ante colisiones; la construcción de una estación de carga inalámbrica Qi, que logró una recarga en promedio de 45–60 minutos con o sin intervención; y la integración de comunicación en tiempo real utilizando Matlab y Python, por medio del uso del protocolo MQTT.

La metodología se ajustó a un enfoque iterativo que incluyó modelado paramétrico, fabricación mediante manufactura digital rápida y validación experimental en vuelos con y sin FlowDeck. En las pruebas, se pudo observar una reducción del 100 % en los daños estructurales tras la instalación de guardas y, también, vuelos autónomos estables utilizando únicamente posicionamiento externo con un error promedio de trayectoria registrado de 3–5 cm.

Los resultados demostraron vuelos más seguros, mayor autonomía gracias a la carga inalámbrica y reducción de riesgos durante la operación, validando la efectividad de la infraestructura propuesta. Este trabajo sienta las bases para futuras investigaciones en navegación autónoma.

Keywords: dron CrazyFlie 2.1, Robotat, seguridad de vuelo, carga inalámbrica.

This paper documents the development and integration of an experimental infrastructure designed to improve the efficiency, stability, and safety of the CrazyFlie 2.1 drone within the Robotat ecosystem at the University of the Valley of Guatemala. The research was conducted in three main phases: the design and optimization of the protective guards, which reduced their mass to 7 g, improving the drone's takeoff capabilities and minimizing damage from collisions; the construction of a Qi wireless charging station, achieving an average recharge time of 45–60 minutes with or without intervention; and the integration of real-time communication using MATLAB and Python, via the MQTT protocol.

The methodology followed an iterative approach that included parametric modeling, rapid digital manufacturing, and experimental validation through flights with and without a FlowDeck. Tests showed a 100% reduction in structural damage after the installation of guards, as well as stable autonomous flights using only external positioning, with an average trajectory error of 3–5 cm.

The results demonstrated safer flights, greater autonomy thanks to wireless charging, and reduced operational risks, validating the effectiveness of the proposed infrastructure. This work lays the groundwork for future research in autonomous navigation.

Keywords: CrazyFlie 2.1 Drone, Robotat, Flight Safety, Wireless Charging.

El Crazyflie 2.1, desarrollado por la empresa Bitcraze, es un microdron de código abierto que se ha convertido en un referente para experimentos en laboratorios de robótica. Su diseño compacto y la posibilidad de programarlo mediante distintos entornos lo convierten en una herramienta versátil para la implementación y validación de algoritmos de control, navegación y localización en espacios interiores.

Sin embargo, el uso del dron CrazyFlie 2.1 en entornos de investigación, particularmente dentro del ecosistema Robotat de la Universidad del Valle de Guatemala, presenta algunas limitaciones que afectan su desempeño experimental. Entre las principales se identifican las siguientes:

- La baja autonomía energética, que obliga a detener los experimentos con frecuencia para recargar manualmente la batería.
- La ausencia de sistemas de protección física, lo cual incrementa el riesgo de daños al dron y a las personas en caso de colisiones.
- La inestabilidad en la comunicación inalámbrica, que puede provocar pérdida de paquetes de datos y afectar la seguridad de los vuelos.

En la Universidad del Valle de Guatemala, el ecosistema Robotat ha sido diseñado como un entorno de experimentación para robots móviles y drones, basado en sistemas de captura de movimiento OptiTrack y comunicación inalámbrica. Si bien construye un espacio idóneo para la validación de algoritmos de control y navegación, las limitaciones mencionadas del CrazyFlie restringen el aprovechamiento pleno de sus capacidades en este ecosistema.

El presente trabajo tiene como finalidad desarrollar infraestructura experimental que incremente la eficiencia y seguridad del dron CrazyFlie 2.1 en el Robotat. Para ello, se plantean tres ejes principales:

- La incorporación de guardas de protección para reducir riesgos en la experimentación.

- La integración de estaciones de carga inalámbrica para extender la autonomía del dron.
- La optimización de herramientas en Python/Matlab para mejorar la comunicación y estabilidad del sistema.

De esta manera, el problema central fue la falta de una infraestructura experimental que garantiza operaciones seguras, continuas y eficientes durante pruebas con drones CrazyFlie 2.1 individuales. Mediante un enfoque de diseño iterativo, se desarrollaron soluciones físicas y digitales que le permiten al dron realizar vuelos autónomos, recargarse de forma inalámbrica y operar con mayor estabilidad dentro del ecosistema Robotat.

En cuanto a la estructura del documento, el contenido se organiza de la siguiente manera:

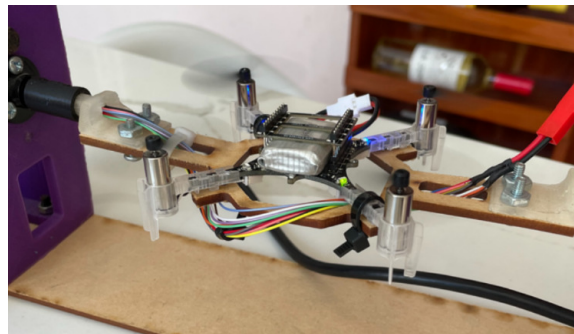
- Los capítulos 2 a 5 presentan los antecedentes, la justificación, los objetivos y la definición del problema.
- El capítulo 6 desarrolla el marco teórico, abordando los componentes del dron CrazyFlie 2.1 y del Ecosistema Robotat.
- Los capítulos 7, 8 y 9 describen el diseño y la implementación de los elementos físicos, como las guardas y las estaciones de carga inalámbrica, así como el proceso de calibración de los markers en el Robotat.
- El capítulo 10 detalla la optimización de las herramientas de software, la integración con Python/Matlab, y las pruebas experimentales de vuelo autónomo hacia la estación de carga.
- Finalmente, los capítulos 11 y 12 presentan las conclusiones y recomendaciones derivadas del trabajo.

En conjunto, este proyecto busca fortalecer las capacidades experimentales del laboratorio de robótica aérea, facilitando la enseñanza, la investigación y la validación de algoritmos de control y navegación autónoma mediante el uso del dron CrazyFlie 2.1.

Los Drones CrazyFlie [1] son plataformas de vuelo de pequeño tamaño desarrolladas por Bitcraze, diseñadas principalmente para aplicaciones de investigación y desarrollo en las áreas de robótica y control. En la Universidad del Valle de Guatemala se han llevado a cabo diversos estudios que exploran el control, la estabilidad y la navegación de estos drones, utilizando distintas herramientas y metodologías.

Inicialmente, Sanabria [2] implementó una plataforma de pruebas para los drones Crazyflie 2.0 en la Figura 1 con el propósito de verificar algoritmos de control de actitud en un grado de libertad. La comunicación con el dron se realizó con Python y el módulo Crazyradio, facilitando la evaluación del comportamiento del dron en condiciones controladas.

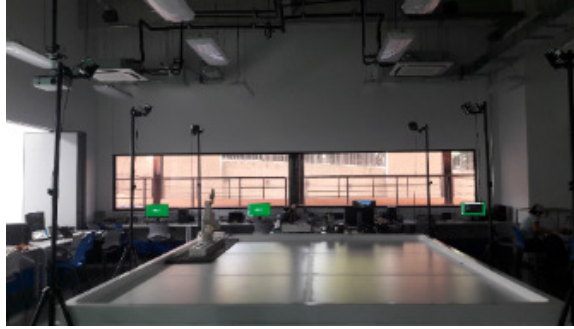
Figura 1. Plataforma de pruebas para drones Crazyflie 2.0



Nota. La imagen muestra la plataforma de pruebas utilizada para experimentación con drones CrazyFlie 2.0. Diseñada para pruebas en entornos controlados y validación de algoritmos [2].

Años después, con el objetivo de brindar apoyo a aplicaciones de robótica autónoma, Perafán [3] presentó el Robotat, se observa en la Figura 2 un ecosistema de experimentación robótica que integra un sistema de captura de movimiento OptiTrack con una red de comunicación Wi-Fi basada en comunicación TCP/IP.

Figura 2. Vista frontal del ecosistema Robotat



Nota. Plataforma Robotat para pruebas con drones CrazyFlie 2.1 en entornos controlados [3].

Posteriormente, Gordillo[4] y Otzoy [5] desarrollaron infraestructura y herramientas para integrar a los Crazyflie dentro del ecosistema Robotat. Se enfocaron en la captura de movimiento del dron, sin utilizar librerías especializadas como Crazyswarm [6] , sino más bien con las herramientas del Robotat directamente. Esto para evitar las dificultades técnicas de emplear un sistema con ROS2 en Ubuntu. Sin embargo, esta decisión probó ser la equivocada ya que no fue posible obtener resultados satisfactorios en ninguno de los dos casos. En adición, Garrido [7] y Ávila [8] continuaron con la misma línea de investigación buscando integrar la librería CrazySwarm en la plataforma de pruebas Robotat con el sistema operativo ROS2. Ávila trabajó en la creación de un servidor que permitiera la comunicación entre drones, con el objetivo de controlar trayectorias tanto de enjambre como individual. Garrido, por otro lado, se enfocó en la implementación de infraestructura para el control y experimentación de múltiples drones mediante ROS2 y Linux, siempre dentro del ecosistema Robotat. Los resultados obtenidos en ambas investigaciones evidenciaron mejoras en la precisión y estabilidad del control de drones, estableciendo una base sólida para futuras investigaciones.

Por último, Pablo Caal [9] empleó un CrazyFlie 2.1 con el sistema LPS (Local Positioning System) para mejorar la navegación mediante el uso del FlowDeck, basado en Odometría visual. El sistema LPS es una alternativa al GPS diseñada para interiores, que permite conocer la precisión del dron dentro de un entorno controlado. El estudio que realizó se enfocó en el seguimiento de trayectorias y en el vuelo autónomo en entornos con obstáculos con el Robotat.

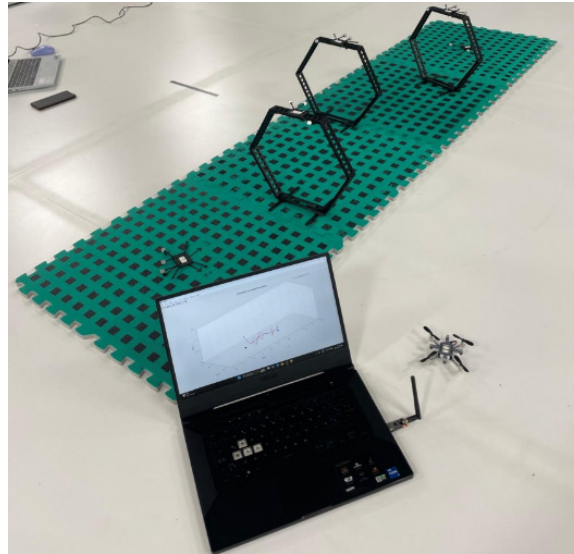
Esto se hizo con el propósito de desarrollar herramientas en software que permitieran su uso sencillo en MATLAB, pero que esto implicó limitaciones en el rendimiento de vuelo del dron, tal como se observa en la Figura 3

Estos estudios han permitido avances significativos en la navegación y control de los drones CrazyFlie, pero aún existe la necesidad de desarrollar una infraestructura experimental que garantice una mejor eficiencia y estabilidad en el vuelo.

En este contexto, el trabajo de Mañas-Álvarez et. al [10], aborda la optimización del control de altitud del CrazyFlie 2.1 mediante un enfoque basado en eventos. Este redujo la frecuencia de actualización de la señal de control mediante la implementación de un

generador de eventos que opera con un umbral relativo. Los resultados obtenidos en la investigación demuestran que esta es una estrategia que mejora la eficiencia del sistema, disminuyendo la cantidad de señales enviadas sin afectar la estabilidad del vuelo. Esto ayuda a que el dron se pueda adaptar muy rápido a cambios en su entorno, especialmente en situaciones que requieren maniobras rápidas, como en carreras de drones, operaciones en espacios reducidos o cambios rápidos de altitud en aplicaciones de rescate.

Figura 3. Pruebas de laboratorio con el dron CrazyFlie 2.1



Nota. Ensayos de vuelo del dron CrazyFlie 2.1 realizados en el laboratorio utilizando el ecosistema Robotat, enfocados en validar mejoras de estabilidad y seguridad [9].

A lo largo de distintas investigaciones en la Universidad del Valle de Guatemala, se han implementado mejoras en el control, estabilidad y navegación del dron CrazyFlie 2.1, utilizando herramientas como ROS2, CrazySwarm y sistemas de captura de movimiento. Sin embargo aún existen limitaciones en la seguridad y eficiencia del sistema.

Unos de los principales desafíos es la implementación de un entorno seguro para la experimentación con drones individuales. La falta de estructuras de protección y estaciones de carga limita la capacidad de realizar pruebas prolongadas sin riesgos de daño al equipo o a las personas que se encuentran alrededor. La optimización de las herramientas de software para la comunicación y manejo del dron sigue siendo un área de mejora, dado que la estabilidad en la transmisión de datos es crucial para garantizar vuelos autónomos precisos.

Este trabajo de graduación busca optimizar el rendimiento y la capacidad de navegación segura el dron CrazyFlie dentro del ecosistema Robotat con tres puntos: la mejora de la infraestructura experimental con guardas de protección, la integración de estaciones de carga inalámbrica para extender la autonomía de los drones y la optimización de software para mejorar la comunicación y control del sistema.

La implementación de estas mejoras traerá beneficios tanto futuros como a la comunidad académica en general. En particular, permitirá integrar el uso de drones en prácticas de laboratorio de forma sencilla y segura, facilitando el desarrollo de nuevas aplicaciones en el área de la robótica. Este trabajo sentará las bases para futuras investigaciones enfocadas en el uso de drones en entornos más complejos.

4.1. Objetivo general

Optimizar el rendimiento y la capacidad de navegación segura de drones CrazyFlie individuales dentro del ecosistema Robotat.

4.2. Objetivos específicos

- Optimizar las herramientas de software para la comunicación y manejo del dron CrazyFlie.
- Diseñar e implementar guardas de protección para el dron CrazyFlie utilizando técnicas de manufactura digital rápida.
- Integrar y utilizar estaciones de carga inalámbrica para el dron CrazyFlie basadas en tecnología Qi de carga.

5.1. Declaración del problema

El presente estudio parte del problema de cómo mejorar la eficiencia de vuelo y la seguridad operativa del dron CrazyFlie 2.1 dentro del ecosistema Robotat mediante la integración de infraestructura experimental adicional. Esta infraestructura combina hardware, como las guardas protectoras y la estación de carga inalámbrica Qi con herramientas de software desarrolladas en MATLAB y Python, proponiendo que su incorporación conjunta incrementa la estabilidad del dron, reduzca los incidentes durante las pruebas y permita una operación más continua y autónoma.

5.2. Relación entre variables

La intervención principal del estudio es la infraestructura experimental mejorada, la cual actúa como variable independiente. Esta comprende, por un lado, los componentes físicos, guardas de protección y estación de carga inalámbrica y, por otro, la integración de software en MATLAB y Python para la comunicación y control del dron. Los efectos de dicha intervención se reflejan en dos variables dependientes: la eficiencia de vuelo, entendida como el tiempo efectivo de operación, continuidad de las pruebas y consumo energético; y la seguridad de operación, asociada a la estabilidad de las trayectorias, la reducción de incidentes y la confiabilidad en la comunicación inalámbrica. La interacción entre estas variables permite evaluar de manera integral cómo la infraestructura propuesta incide en el desempeño del dron dentro del entorno Robotat.

5.3. Comprobación y cuantificación

La naturaleza del problema permite una validación objetiva, ya que los cambios generados por la infraestructura pueden medirse experimentalmente. El incremento en el tiempo de vuelo, la disminución de interrupciones por recargas manuales, la reducción de fallos en la comunicación y la ausencia de colisiones o daños durante las pruebas constituyen indicadores observables y cuantificables. Estos parámetros permiten comparar el estado inicial del sistema con los resultados obtenidos tras la implementación, verificando si la infraestructura experimental mejorada cumple con los objetivos planteados.

5.4. Delimitación del estudio

El problema se enfoca en la validación experimental de drones CrazyFlie 2.1 individuales dentro del ecosistema Robotat, en condiciones interiores controladas. Se excluyen de este análisis drones de mayor escala, entornos exteriores o aplicaciones comerciales, ya que el objetivo principal es fortalecer la capacidad académica en robótica aérea.

Con el fin de cumplir los objetivos planteados en este proyecto, es fundamental desarrollar un marco teórico que aborde los principales conceptos y componentes relacionados con el dron Crazyflie 2.1. En este apartado se describirá qué es y cómo funciona el dron Crazyflie 2.1. Además, se tratarán aspectos relacionados con la seguridad del dron, especialmente en lo que respecta a la protección de las hélices, y se explicará el funcionamiento de los cargadores inalámbricos Qi. Finalmente, se explorará la estructura y funcionalidad del software `cfclient` y el uso de Python/MATLAB para el control y análisis del dron Crazyflie 2.1.

6.1. CrazyFlie 2.1

El Crazyflie 2.1 [11], es un microdron de código abierto, destacado por su diseño compacto y bajo peso, con tan solo 27 gramos, lo que le permite caber en la palma de la mano, como se muestra en la Figura 4. A pesar de su reducido tamaño, ofrece capacidades avanzadas de comunicación mediante Bluetooth Low Energy, una radio de baja latencia y largo alcance. Esto permite controlar el dron desde un dispositivo móvil a través de una aplicación, o desde un computador utilizando un módulo Crazyradio PA o Crazyradio.

Figura 4. Dron CrazyFlie 2.1



Nota. El dron crazyflie 2.1 es un microdron de código abierto que se usa para el investigación y desarrollo [11].

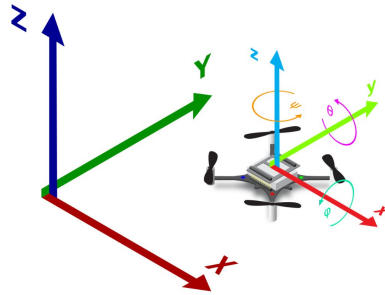
6.1.1. Sistema de ejes de un cuadricóptero

El vuelo de un cuadricóptero como el Crazyflie 2.1 se basa en el control coordinado de cuatro variables fundamentales: *roll*, *pitch*, *yaw* y *thrust* como se muestra en la Figura 6. Estas variables definen la actitud y la altitud del dron durante el vuelo, y son controladas mediante una interfaz, ya sea una aplicación móvil, entre otros. Estos controles permiten maniobrar el dron en el espacio tridimensional y son esenciales tanto para el vuelo manual como para la implementación de estrategias de control automático.

El sistema de coordenadas que usan en la familia de drones Crazyflie 2.1 sigue la convención ENU (Este-Norte-Superior) tanto para el sistema global como el sistema de referencia o cuerpo. Por lo tanto, los ejes globales \mathbf{X} , \mathbf{Y} , \mathbf{Z} y los ejes del cuerpo x , y , z son orientados de la siguiente forma:

- El eje X sigue apuntando hacia Este.
- El eje Y sigue apuntando hacia el Norte.
- El eje Z sigue apuntando hacia arriba. Se puede observar mejor en la Figura 5.

Figura 5. Sistema de ejes de un cuadricóptero



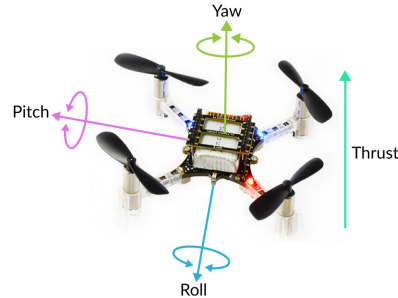
Nota. El diagrama ilustrativo del sistema de ejes ENU es utilizado en el CrazyFlie 2.1, mostrando la orientación de los ejes globales y del cuerpo, fundamentales para el control de navegación [12].

En cuanto a las rotaciones que se muestran en la Figura 6, los ángulos de actitud *yaw*, *pitch*, *roll* siguen las a continuación mencionadas reglas de rotación:

- *Roll*: corresponde a la rotación alrededor del eje X del dron. Un giro positivo se produce cuando el dron rota en sentido horario visto desde la cola hacia el frente, es decir, cuando inclina el ala derecha hacia abajo.
- *Yaw*: se define como la rotación alrededor del eje Z, que determina la orientación o dirección del dron. Un giro positivo implica una rotación en sentido horario visto desde arriba, también conforme a la regla de la mano derecha.
- *Pitch*: representa la rotación alrededor del eje Y. Un giro positivo ocurre cuando el dron desciende el morro (frente) y eleva la parte trasera, siguiendo la regla de la mano derecha alrededor del eje Y.

Esta convención es importante para la correcta interpretación de los datos de posición del dron tanto en las simulaciones como en la implementación del control [12].

Figura 6. Convenciones de rotación para los ángulos de actitud



Nota. *Yaw* y *roll* giran según la regla de la mano derecha, *pitch* según la mano izquierda. Esto es importante para interpretar la orientación del dron [12].

6.1.2. Crazyradio PA

Crazyradio PA [13] es un adaptador USB de radiofrecuencia de código abierto y largo alcance, basado en el chip nRF24LU1+ de Nordic Semiconductor, como se muestra en la Figura 7. Incorpora un amplificador de potencia de 20 dBm y un LNA y viene con firmware preinstalado compatible con Crazyflie. Gracias al amplificador, puede alcanzar distancias de hasta 1 km cuando se usa con Crazyflie 2.1 y hasta 2 km entre dos dispositivos Crazyradio PA, siempre bajo condiciones ideales.

Figura 7. Comunicación inalámbrica del dron CrazyFlie 2.1



Nota. El Crazyflie 2.1 se comunica de forma inalámbrica mediante el módulo Crazyradio PA, permitiendo control de baja latencia y largo alcance [13].

Al tratarse de un proyecto de código abierto, con firmware desarrollado desde cero y una API en Python para su control, es una opción ideal para sistemas que requieren mayor alcance que el WiFi, sin necesidad de un gran ancho de banda.

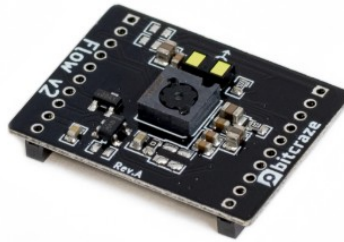
6.1.3. Flow Deck

El Flow Deck [14] es una expansión para el Crazyflie 2.1 que le permite detectar su propio movimiento en cualquier dirección. Esto se logra mediante el uso de dos sensores:

- Un sensor de tiempo de vuelo (ToF) VL53L1x, que mide con alta precisión la distancia al suelo.
- Un sensor de flujo óptico PMW3901, que registra el desplazamiento relativo respecto a la superficie.

La combinación de ambos permite al dron mantener una estimación confiable de su posición en vuelo, facilitando tanto la navegación autónoma en entornos tridimensionales como la estabilización del vuelo. El módulo como tal se muestra en la Figura 8.

Figura 8. Placa de expansión Flow Deck



Nota. La placa Flow Deck integra un sensor óptico y un sensor de distancia, permitiendo al CrazyFlie 2.1 medir velocidad relativa y altura para un control de vuelo más preciso [14].

6.1.4. Biblioteca para Python de Crazyflie

La biblioteca para Python de Crazyflie (**CFLib**) [15] ha sido diseñada para el dron Crazyflie 2.1. Es capaz de automatizar una amplia gama de tareas, como volar el dron, leer valores de sensores, gestionar parámetros internos, ejecutar scripts y muchas otras en una interfaz de programación de alto nivel fue desarrollada por Bitcraze. A continuación se describen sus componentes principales.

CFLib de Crazyflie permite un enlace inalámbrico con el Crazyflie que puede establecerse utilizando un Crazyradio PA. Esta comunicación es confiable, lo que facilita la detección automática del dron, la reconexión tras pérdidas temporales de enlace y la gestión simultánea de múltiples dispositivos. Dentro de la librería, la clase `Crazyflie` y el administrador de enlaces `SyncCrazyflie` simplifican la inicialización de la comunicación y el envío de comandos, permitiendo al usuario establecer, mantener y controlar las conexiones de manera programática desde Python.

En la librería CFLib, todos los canales de comunicación con el dron se identifican utilizando una dirección URI, que es una cadena con un formato estructurado utilizada para

especificar de manera única el tipo de enlace, el puerto, la velocidad de comunicación y la ruta utilizada para conectar con el dron con un formato estructurado que especifica el tipo de:

- Interfaz, su identificador, el canal de comunicación y en algunos casos, la velocidad de transmisión.

Por ejemplo: Una URI como **radio://0/10/2M** representa una conexión vía radio a través del dongle USB número 0, utilizando el canal 10 y una velocidad de 2 Mbps.

Otros tipos de interfaz disponibles incluyen conexiones de:

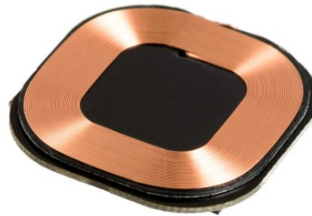
- **Depuración:**
 - `debug://0/1`
- **Conexión USB directa al microcontrolador:**
 - `usb://0`
- **Puertos seriales:**
 - `serial://ttyAMA0`
- **Enlaces TCP/IP:**
 - `tcp://tcp-1.local:300`

Esta estructura estandarizada permite que el sistema reconozca y configure automáticamente la conexión adecuada para comunicarse con el dron [15].

6.1.5. Qi 1.2 *charger deck*

La placa de expansión de carga inductiva Qi 1.2 [16] permite recargar el Crazyflie sin necesidad de conectar el cable micro USB, ofreciendo una solución inalámbrica práctica. Compatible con el estándar Qi WPC versión 1.2, esta placa funciona con los cargadores Qi comunes para teléfonos y tablets, el módulo como tal se observa en la Figura 9.

Figura 9. Qi 1.2 charger deck



Nota. La placa Qi 1.2 Charger Deck permite la carga inalámbrica del CrazyFlie 2.1 mediante el estándar Qi WPC 1.2, eliminando la necesidad de cables y facilitando la operación continua [16].

Su diseño eléctrico se basa en el circuito integrado TI BQ51013B, que asegura una carga inalámbrica segura. Puede suministrar hasta 1A a 5V, garantizando una carga eficiente

para el Crazyflie. Además, cuenta con detección automática mediante memoria 1-wire para identificar la placa de expansión conectada. Con un tamaño compacto de $30 \times 30 \times 5$ mm y un peso de 5 gramos, está diseñada específicamente para montarse bajo el Crazyflie 2.1.

6.2. Cargador inalámbrico Tozo W1

El cargador inalámbrico TOZO W1 [17] es compatible con diversos dispositivos móviles, incluyendo modelos recientes de iPhone, Samsung Galaxy y audífonos como los AirPods. Utiliza el estándar Qi y ofrece una carga rápida con potencias de hasta 15 W, permitiendo una recarga eficiente en menor tiempo. Se puede observar en la Figura 9 que tiene un diámetro de 100 mm y un grosor de 5 mm.

Figura 10. Tozo cargador inalámbrico actualizado 2025



Nota. Cargador inalámbrico TOZO 2025 es compatible con el estándar Qi, utilizando para recargar la placa Qi 1.2 Charger Deck del CrazyFlie 2.1 [17].

Además, incorpora protección contra sobrecarga, sobretensión y cortocircuitos, así como un puerto USB tipo C reversible. Su diseño compacto incluye materiales antideslizantes para mantener el dispositivo estable durante la carga y un indicador LED discreto que no interfiere con el entorno.

6.3. Motion capture marker deck

La expansión *Motion Capture Marker Deck* [18] permite montar de forma sencilla marcadores reflectantes en el dron CrazyFlie, como se observa en la Figura 11, facilitando su seguimiento dentro de un sistema de captura de movimiento. Los marcadores son detectados por las cámaras del sistema, que calculan la posición exacta del dron. Esta información puede emplearse para mediciones precisas o para enviar retroalimentación al CrazyFlie y habilitar vuelos autónomos.

Los *Motion Capture Marker Deck* [19] son elementos que no generan luz propia, sino que están diseñados para reflejar la luz infrarroja emitida por las cámaras del sistema de captura. Estas cámaras proyectan haces de luz IR que inciden sobre los marcadores reflectantes

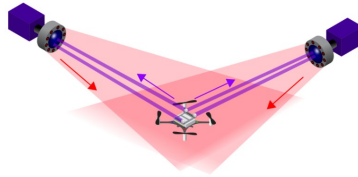
Figura 11. *Motion capture marker deck*



Nota. El *Motion Capture Marker Deck* facilita la detección precisa de la posición del dron mediante cámaras en sistemas de captura de movimiento [18].

montados en el dron. La luz reflejada regresa a las cámaras, que detectan su posición en el espacio, como se observa en la Figura 12.

Figura 12. *Passive marker deck* reflectantes utilizados en sistemas de captura de movimiento



Nota. Los *Passive Marker Deck* reflejan la luz infrarroja emitida por las cámaras, permitiendo la detección precisa de la posición del dron mediante triangulación [20].

Mediante técnicas de triangulación basadas en la información obtenida por múltiples cámaras, el sistema calcula con alta precisión la posición y orientación tridimensional del dron en tiempo real. Esto permite generar datos fiables para controlar y monitorear el vuelo del CrazyFlie.

La información de posición y orientación obtenida por los marcadores pasivos se puede usar para enviar retroalimentación al controlador del dron, habilitando modos de vuelo autónomo, estabilización o seguimiento de trayectorias [20]. Además, facilita la captura de datos para investigaciones y pruebas en ambientes controlados.

Diseño de guardas para hélices del dron Crazyflie 2.1

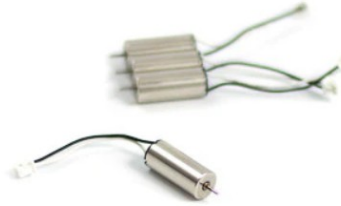
Este capítulo presenta el proceso completo de diseño, iteración y validación de las guardas de protección para las hélices del dron CrazyFlie 2.1. Se detalla cómo las dimensiones del dron, motores y hélices condicionaron el modelado paramétrico en Autodesk Inventor, y cómo distintos prototipos fueron evaluados estructuralmente estable. El objetivo principal es documentar las decisiones técnicas que permitieron desarrollar una guarda capaz de mejorar la seguridad operativa del dron sin comprometer su capacidad de vuelo.

7.1. Enfoque general de diseño

El objetivo de las guardas protectoras fue reducir el riesgo de daño al dron durante vuelos experimentales en el ecosistema Robotat, sin comprometer la estabilidad ni la capacidad de empuje. Para ello se desarrolló un proceso de diseño basado en iteraciones sucesivas, evaluando distintos parámetros geométricos y estructurales hasta obtener una solución funcional, ligera y adaptable al CrazyFlie 2.1.

El diseño de las guardas se basó en tres factores principales: la geometría del dron, incluyendo la distancia entre motores y la separación entre hélices, las dimensiones de los motores y soportes (tanto los originales de 4x7 mm sin núcleo de CC, mostrados en la Figura 13, como los modelos de mayor empuje), y las características de las hélices, cuyo diámetro y paso determinan el radio mínimo que debía protegerse. Estos parámetros establecieron las restricciones geométricas necesarias para el modelado de la guarda y aseguraron que el diseño final fuera compatible con la estructura del CrazyFlie 2.1 sin generar interferencias durante el vuelo.

Figura 13. Motor sin núcleo de CC



Nota. Motor sin núcleo de corriente continua con dimensiones compactas, utilizando en el diseño original del dron [21].

Dimensiones de los nuevos motores:

- 7 x 20mm Motor sin núcleo de CC

Figura 14. Paquete de motor de CC (mejora de empuje)



Nota. Motor sin núcleo de CC con mayor tamaño y capacidad, diseñado para mejorar el empuje del dron [22].

Dimensiones de los soportes del motor: 4 × 7 mm

Figura 15. Soportes del motor



Nota. Dimensiones y diseño de los soportes utilizados para fijar los motores al dron CrazyFlie 2.1 [23].

Las dimensiones de los motores tanto los originales como los de mayor empuje determinaron el diámetro interior mínimo de los soportes para la guarda. Cualquier error de tolerancia en este punto comprometería el montaje o provocaría vibraciones durante el vuelo.

La hélice 47-17 (4CCW+4CW), mostrada en la Figura 16, utilizada en el CrazyFlie 2.1

cuenta con un diámetro de 47 mm y un paso de 17 mm, está fabricada en policarbonato (PC) y es compatible con ejes de 0.8 mm. Su peso es de 0.3 g y posee dos cuchillas, lo que lo hace adecuada para mantener un equilibrio entre eficiencia aerodinámica y baja masa, ambos factores esenciales para el rendimiento del dron.

Figura 16. Hélice 47-17 (4CCW+4CW) color negro



Nota. Hélice de doble cuchilla con especificaciones de diámetro y paso, diseñada para optimizar el rendimiento del dron CrazyFlie 2.1 [24].

La hélice QProp, mostrada en la Figura 17, empleada en las pruebas presenta un diámetro de 51 mm (2") y un paso de 22.9 mm (0.9"), está fabricada en policarbonato (PC) y cuenta con un radio de cubo de 4 mm. Su peso es de 0.27 g y posee dos cuchillas, características que la convierten en una opción ligera y eficiente para mejorar el desempeño aerodinámico del dron CrazyFlie 2.1.

Figura 17. HQ Ultralight Prop 2X0.9 (51MM) Grey(2CW+2CCW)-PC-1MM Shaft



Nota. Hélice QProp ligera de dos cuchillas, diseñada para eficiencia y compatibilidad con ejes de 1 mm [25].

Las dimensiones de las hélices fueron igualmente críticas. El radio y el paso de cada modelo de hélice definieron el espacio que debía rodearse con el aro de protección. Para garantizar que no existiera contacto entre el aro y las hélices durante maniobras, se estableció un margen de seguridad calculado directamente a partir del diámetro real de las hélices mostrado en las Figuras 16 y 17. Este margen influyó directamente en el diámetro total de la guarda.

Medidas esenciales del dron:

- Longitud Total: 56.650 mm
- Ancho: 30.000 mm

deficiencias mostraron que el diseño no cumplía con los requisitos de protección, peso y desempeño de vuelo, por lo que se descartó y se procedió a desarrollar una nueva iteración utilizando TPU, un material más ligero y flexible que se ajusta mejor a las necesidades del sistema.

Figura 19. Primer diseño

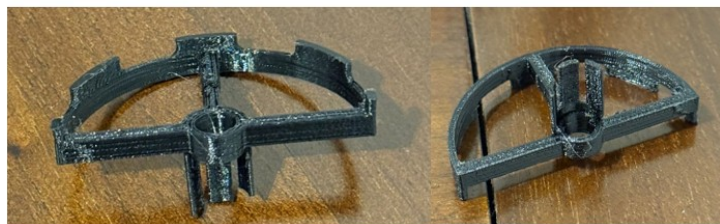


Nota. Primer diseño de guarda impreso en PLA, con limitaciones en peso y protección para el dron CrazyFlie 2.1. Elaboración propia.

El segundo diseño de guarda protectora:

El segundo diseño que se encuentra en la Figura 20 fue fabricado en TPU, lo cual representó una mejora respecto al prototipo inicial, ya que redujo el peso y permitió un despegue parcial. Sin embargo, durante la evaluación según los criterios de seguridad y rigidez, el dron tendía a volcar poco después de elevarse. Asimismo, el mecanismo de sujeción basado en tres patas delgadas no cumplió con los requisitos de resistencia mecánica: las patas se desprendían con facilidad y presentaban fracturas, lo que impedía una fijación adecuada al chasis del CrazyFlie 2.1. Debido a estos problemas, este diseño fue descartado para continuar con una siguiente iteración.

Figura 20. Segundo diseño



Nota. Segundo diseño impreso en TPU, con mejor ligereza, pero tiene problemas de sujeción y estabilidad en el vuelo. Elaboración propia.

El tercer diseño de guarda protectora:

En el tercer diseño Figura 21, se eliminó una de las tres patas del sistema de sujeción con el fin de mejorar el flujo de aire alrededor de las hélices y aumentar la estabilidad del dron. Además, se reforzó la unión con un soporte adicional para reducir la flexión de las varillas. A pesar de estas modificaciones, el rendimiento del dron seguía siendo insuficiente, según los criterios de empuje y seguridad. Esto evidenció que la limitación no provenía únicamente del diseño de la guarda, sino del conjunto motriz. Por esta razón, se procedió a reemplazar los motores y hélices por versiones de mayor potencia, lo que finalmente permitió que el dron despegara de manera adecuada y segura.

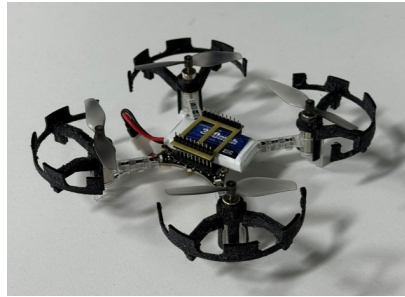
Figura 21. Tercer diseño



Nota. Tercer diseño con modificaciones en soporte y estructura para mejorar la estabilidad y el flujo de aire, previo a la actualización de motores y hélices. Elaboración propia.

Luego, se realizaron pruebas de vuelo para evaluar el impacto del peso de la guarda final sobre el desempeño del CrazyFlie 2.1. El dron logró elevarse sin dificultades de forma segura durante maniobras básicas, confirmando que el diseño cumplía con los requisitos de empuje, rigidez y margen de seguridad alrededor de las hélices. Además, se verificó que la estructura proporciona protección efectiva ante colisiones tanto con objetos como con personas Figura 22.

Figura 22. Dron CrazyFlie 2.1 con guardas instaladas



Nota. Las guardas instaladas reducen daños por impactos y fueron validadas en pruebas de vuelo, comprobando que el peso adicional no afecta la capacidad de despegue ni la estabilidad. Elaboración propia.

Una vez validado el diseño y confirmada su compatibilidad con el dron, las guardas fueron instaladas en diez drones adicionales en la Figura 23 utilizando en un proyecto paralelo, con el propósito de reducir el riesgo de daño durante sus vuelos.

Figura 23. Diez drones CrazyFlie 2.1 con guardas instaladas

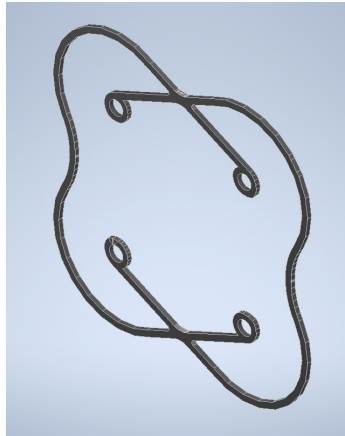


Nota. Diez drones CrazyFlie 2.1 con guardas instaladas para protegerlos durante su uso en proyectos adicionales, mostrando la preparación de los equipos antes de operarlos. Elaboración propia.

En las primeras pruebas, las guardas diseñadas cumplieron correctamente su función de protección, permitiendo realizar vuelos estables y sin incidentes. Sin embargo, tras varios días de operación se observaron algunas fallas relacionadas principalmente con el peso y la distribución del mismo.

El exceso de masa en cada brazo del dron generaba un leve desbalance que afectaba la estabilidad durante maniobras rápidas o incluso durante en el despegue y prolongaba el tiempo de respuesta del control de actitud. Ante esto, se decidió cambiar el diseño de las guardas considerando un enfoque más liviano, como en la Figura 24.

Figura 24. Diseño de la guarda de protección para las hélices

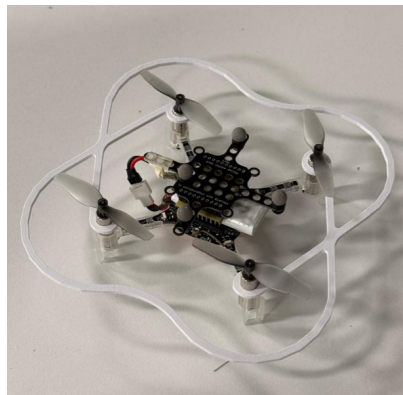


Nota. Vista general del diseño individual de la guarda de protección. Esta versión inicial fue fabricada en material PLA y posteriormente optimizada para reducir peso y mejorar la estabilidad del dron [27].

El diseño inicial de las guardas fue tomado de un modelo de referencia disponible en línea [27], el cual sirvió como base para la implementación. A partir de este diseño, se realizaron modificaciones específicas en las dimensiones del radio de sujeción y en la zona de acople a los motores, con el objetivo de lograr un ajuste más preciso y seguro en la estructura del dron.

En comparación con las guardas iniciales, las nuevas presentaron con un peso de 7g y una mejora notable en la estabilidad dinámica convirtiéndose en la versión final implementada en el dron CrazyFlie 2.1, en la Figura 25.

Figura 25. Dron con guarda integrada



Nota. Ensamble del dron con las guardas de protección integradas. El diseño final permitió una reducción de peso y mejora notable en la estabilidad durante el vuelo. Elaboración propia.

Diseño e implementación de estaciones de carga para drones CrazyFlie 2.1

Con el diseño y validación de las guardas protectoras para el dron CrazyFlie 2.1, el siguiente paso del proyecto consistió en desarrollar la infraestructura necesaria para permitir operaciones continuas dentro del ecosistema Robotat. En particular, fue indispensable un almacenamiento seguro, una organización eficiente y un proceso de recarga confiable para los drones durante las pruebas experimentales. Este capítulo presenta el proceso de diseño, iteración y fabricación de los protectores y bases para los cargadores inalámbricos Qi Tozo, describiendo los requisitos funcionales que guiaron su construcción, las alternativas evaluadas y las soluciones finales adoptadas. Con ellos, se establece un seguimiento al objetivo general de adaptar físicamente el hardware del CrazyFlie 2.1 para su uso seguro.

8.1. Requerimientos funcionales del diseño

Durante la etapa inicial del proyecto se definieron los requerimientos funcionales que debían cumplir las estaciones de carga. En particular, el diseño debía permitir la colocación precisa del cargador inalámbrico Qi Tozo, asegurar la posición del dron CrazyFlie 2.1 durante el proceso de carga, proteger los componentes electrónicos de golpes o desgaste por uso continuo y mantener dimensiones compatibles con el espacio de trabajo del ecosistema Robotat. Además, se buscó que la estructura fuera resistente, fácil de manipular y lo suficientemente ligera para no dificultar el transporte y almacenamiento de los drones.

8.2. Diseño de los protectores de los cargadores inalámbricos Tozo

Para el diseño de los cargadores inalámbricos Tozo en la Figura 10, se busca crear un protector resistente que evite daños y facilite su colocación sobre la plataforma Robotat. Con este diseño, se mejoró la organización y se optimizó el tiempo al momento de posicionarlos, haciendo el proceso más ordenado.

8.2.1. Diseño, ajustes realizados en la caja y requisitos de diseño

Para el diseño de la base se debió considerar el tamaño del dron CrazyFlie 2.1, de manera que pueda colocar bien el cargador Tozo, mostrando en la Figura 10, así como al dron completo con sus guardas.

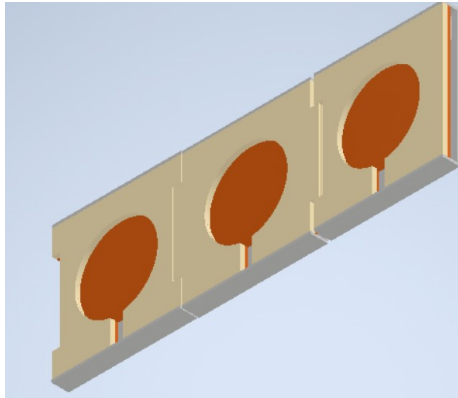
Durante el proceso de diseño se incorporó la geometría del dron CrazyFlie 2.1 y del cargador Qi Tozo como referencia dimensional, permitiendo modelar una base que alineara correctamente ambos componentes y garantizara compatibilidad con las guardas previas del capítulo anterior.

8.3. Proceso de modelado de la caja

Primer diseño:

Utilizando el programa Autodesk Inventor, se planteó inicialmente el diseño de bases tipo estuche para los cargadores Tozo. El objetivo era que, al retirar todos los drones, los estuches pudieran ensamblarse como un rompecabezas, manteniéndose en su lugar y facilitando su colocación sobre cualquier plataforma en la Figura 26. Además, se consideró un canal inferior para que los cables de los cargadores pasaran por debajo de los estuches, quedando ocultos y evitando interferencias visuales y físicas.

Figura 26. Estuche para cargadores Tozo

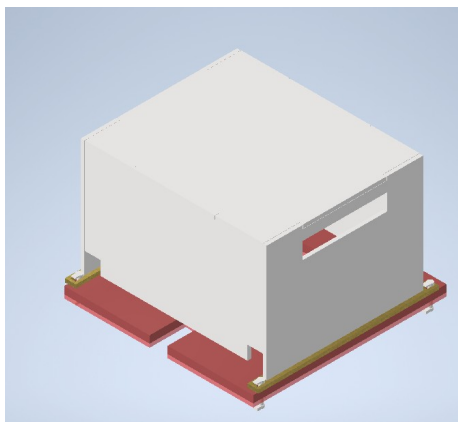


Nota. Modelo del estuche diseñado en Inventor Autodesk, mostrando cómo los cargadores Tozo pueden organizarse de manera compacta y estable dentro de la base. Elaboración propia.

Segundo diseño:

Al ver que el primer diseño de la Figura 26 presentaba algunos problemas como que al ejecutar las estaciones los drones quedaban muy cerca entre sí y podía ser complicado colocarlos correctamente, entonces se decidió cambiar el enfoque. Se optó por hacer cajas individuales que permitan guardar de una vez el dron junto con el cable del cargador, evitando tener que sacar todo por separado. Además, cada dron tendría su propia caja con un número asignado, para que no se confundan al momento de usarlos, como en la Figura 27.

Figura 27. Estación de carga para cargadores Tozo (versión 2)



Nota. Cada dron CrazyFlie 2.1 cuenta con su propia caja numerada, facilitando el orden y la movilización. Elaboración propia.

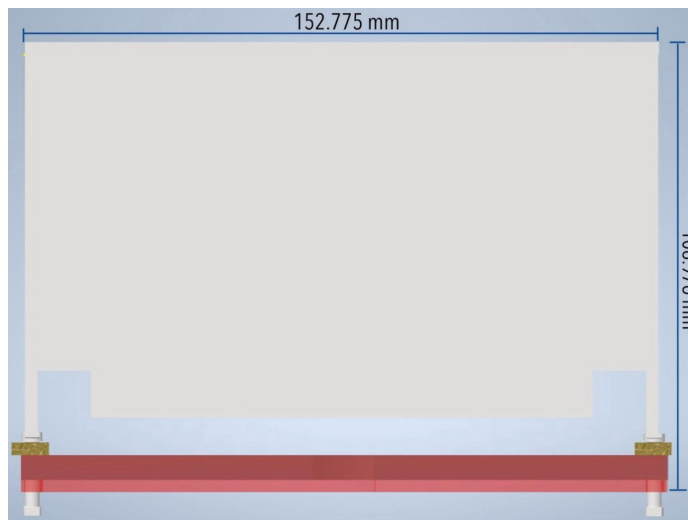
Con el uso de la cortadora láser y la impresora 3D fue posible fabricar y ensamblar físicamente la caja, como en la Figura 28.

Figura 28. Estación de carga física para cargadores Tozo (versión 2)



Nota. Resultado físico del armado de la caja con corte láser e impresión 3D. Elaboración propia.

Figura 29. Representación dimensional de la caja de la estación de carga.



Nota. Representación dimensional de la caja, donde se ilustraron las dimensiones finales obtenidas tras el proceso de diseño y ajuste del modelo. Elaboración propia.

Las dimensiones finales de la caja se sintetizaron mediante una representación dimensional para una presentación más clara y estandarizada. En la Figura 29 se muestran las medidas principales del diseño, como:

Dimensiones de la caja completa:

- Altura de 108.976 mm

- Ancho de 152.775 mm
- Material utilizado fue MDF

Plataforma de estación de carga para el dron Crazyflie 2.1

Antes de desarrollar los distintos prototipos de la estación de carga, se establecieron los requerimientos funcionales que el diseño debía cumplir. En la base de la caja debía integrarse un cargador inalámbrico Qi Tozo, asegurando un soporte estable y correctamente alineado con la posición de carga del dron. Asimismo, el dron CrazyFlie 2.1 debía ubicarse en el centro de la estructura, permitiendo que pudiera cargarse de manera eficiente y simultáneamente almacenarse de forma segura dentro de la caja en la Figura 30. Estos requisitos guiaron todas las iteraciones posteriores del diseño, de acuerdo con la metodología de diseño mecánico propuesta por Norton, en la que los criterios funcionales iniciales gobiernan la generación y evaluación de las alternativas.

Figura 30. Base de estación de carga



Nota. Vista de la base de la caja con la estación de carga Qi Tozo y el dron CrazyFlie 2.1 en el centro. Elaboración propia.

A continuación, se detalla cómo se resolvió el mecanismo de cierre de la caja, desarrollado como parte del proceso iterativo del diseño para garantizar un almacenamiento seguro y evitar aperturas accidentales durante el transporte.

Para cumplir con el requisito de asegurar la tapa sin incrementar significativamente el peso del sistema, se desarrolló un mecanismo de cierre en forma de llave. Este mecanismo permitía accionar simultáneamente la base y la tapa mediante un movimiento rotacional, siguiendo el principio de un pestillo mecánico convencional. Su geometría fue adaptada para encajar de manera precisa en la estructura de la caja, tal como se ilustra en la Figura 31.

Dimensiones de la llave:

- Altura de 20 mm

- Ancho de 7.314 mm
- Radio de 2.57 mm

Material de la llave: Material PLA

Figura 31. Llave para asegurar la caja de estación de carga

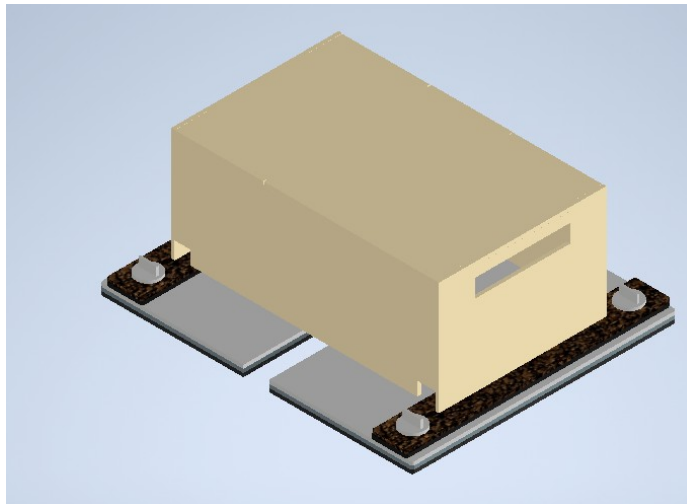


Nota. Vista del mecanismo de cierre tipo llave diseñado para la caja.
Elaboración propia.

Tercer diseño:

Durante la evaluación del segundo diseño en la Figura 28, se identificó que la llave original presentaba un tamaño insuficiente para una manipulación adecuada. Su reducido diámetro dificultaba la operación del cierre, especialmente durante el uso repetitivo esperado en entornos experimentales. Conforme a la metodología de diseño, este hallazgo condujo a iterar hacia un tercer diseño con dimensiones ligeramente mayores, lo cual facilitó el accionamiento del mecanismo y mejoró su confiabilidad operativa, como en la Figura 32.

Figura 32. Estación de carga para cargadores Tozo (versión 3)

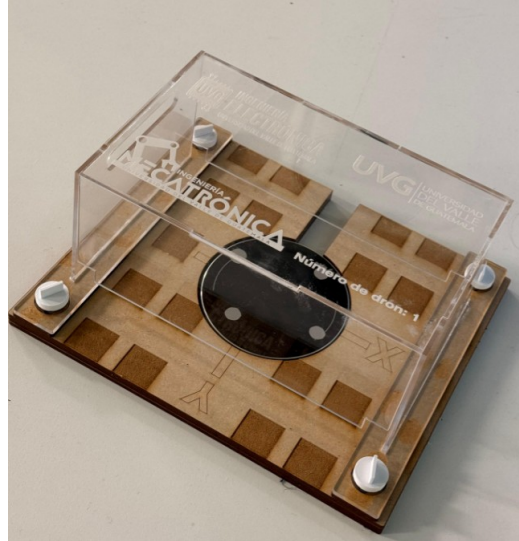


Nota. El segundo diseño presentaba una llave demasiado pequeña, por lo que se planteó ampliar la caja para facilitar su uso. Elaboración propia.

Con el fin de mejorar tanto la estética como la funcionalidad del sistema, la tapa final se

fabricó en acrílico transparente, incorporando los logotipos institucionales de la Universidad del Valle de Guatemala y de las carreras de Mecatrónica y Electrónica. Las llaves fueron impresas en color blanco para garantizar uniformidad visual y facilitar su identificación en el entorno de laboratorio, como en la Figura 33.

Figura 33. Caja completa de la estación de carga final



Nota. Vista general de la caja fabricada en acrílico incluyendo los logotipos institucionales y MDF. Se incluyen los y el patrón cuadrulado en la base para facilitar el funcionamiento del módulo FlowDeck durante el despegue y aterrizaje. Elaboración propia.

La base de la estación de carga fue fabricada en MDF e incluye un patrón cuadrulado diseñado para cumplir con el requerimiento de compatibilidad con el módulo FlowDeck del dron. Este patrón evita atascos y mejora la estabilidad durante el proceso de despegue y aterrizaje, como en la Figura 34.

Figura 34. Estación de carga final con patrón



Nota. Se incluye a la estación de carga el patrón cuadrulado en la base para facilitar el funcionamiento del módulo FlowDeck durante el despegue y aterrizaje. Elaboración propia.

8.4. Calibración del cargador Qi Tozo como marker en Robotat

Antes de realizar las pruebas de vuelo autónomo, fue necesario calibrar el cargador inalámbrico Qi Tozo como un marker dentro del ecosistema Robotat. Esta calibración permitió que la estación de carga pudiera ser detectada correctamente por el sistema de cámaras, asegurando su identificación como un objetivo válido para las maniobras del dron. Además, al asignarle un patrón único y una etiqueta numérica, el Robotat pudo estimar con precisión las coordenadas de la estación de carga, información indispensable para que el CrazyFlie 2.1 ejecutara aproximaciones, aterrizajes y procesos de recarga. Gracias a esta integración, la estación de carga no solo funcionó como un módulo físico, sino como un punto de referencia espacial dentro del entorno de navegación.

8.5. Requerimientos para la generación del patrón de marker en la estación de carga

Para garantizar que la estación de carga pudiera ser detectada por el ecosistema Robotat, fue necesario establecer una serie de requerimientos geométricos y materiales asociados al patrón reflectivo que funcionaría como marker. Estos requerimientos guiaron la construcción del patrón y sustituyeron la necesidad de un procedimiento secuencial explícito. Entre ellos se incluyeron los siguientes:

- Se requirió utilizar cinta adhesiva reflectiva, preferiblemente de color gris Figura 35 para asegurar que las cámaras del Robotat detectaran el patrón con suficiente con-

traste.

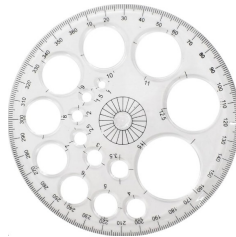
Figura 35. Cinta adhesiva reflectiva



Nota. Uso de cinta adhesiva reflectiva gris para optimizar la detección del marker en Robotat. Elaboración propia.

- Fue necesario emplear una platilla de círculos Figura 36 para obtener recortes homogéneos, garantizando consistencia en la apariencia y el tamaño de los puntos reflectivos colocados sobre el cargador Qi Tozo.

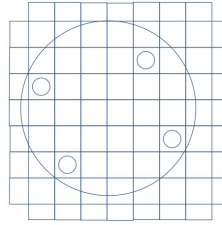
Figura 36. Plantilla de círculos



Nota. Plantilla empleada como guía para obtener recortes circulares del diseño. Elaboración propia.

- Debido a que el sistema debía operar con diez drones CrazyFlie 2.1, se estableció el requerimiento de diseñar diez patrones distintos, permitiendo que el Robotat identificara cada estación de carga de manera individual.
- Para definir la ubicación exacta de cada punto reflectivo, se requirió construir una cuadrícula compuesta por siete líneas verticales y seis horizontales, espaciadas uniformemente a 15 mm. Esta cuadrícula aseguró precisión en la geometría del patrón, como se muestra en la Figura 37.

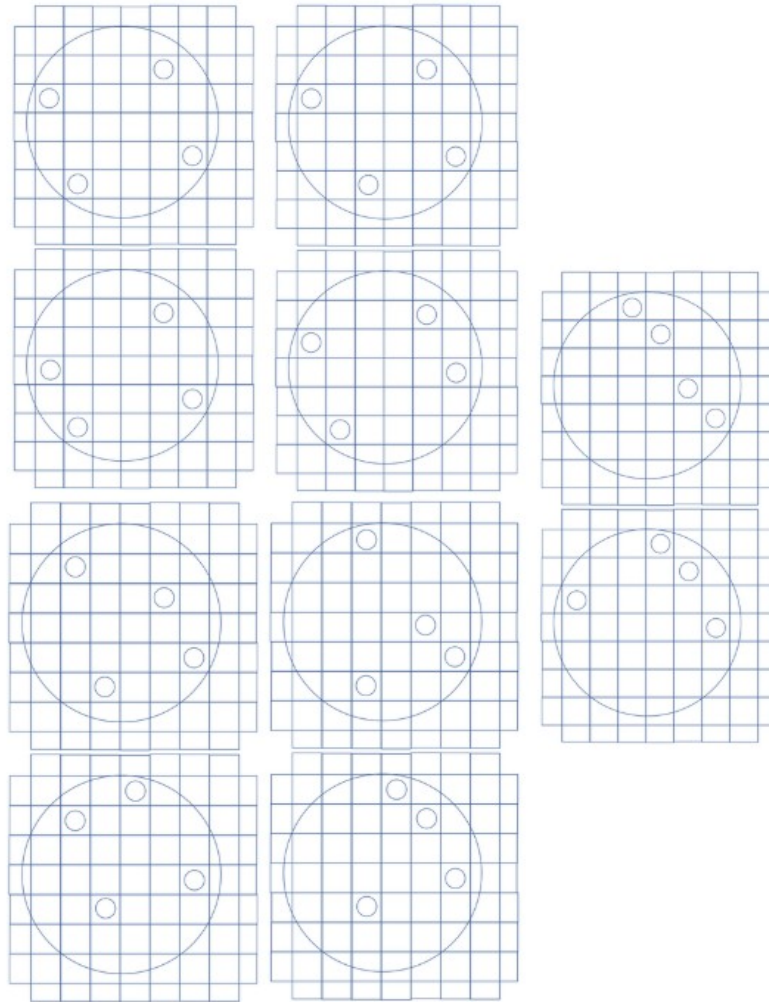
Figura 37. Patrón para estación de carga



Nota. Distribución de líneas poder seleccionar que tipo de patrones utilizar para generar markers. Elaboración propia.

- El diámetro de cada círculo reflectivo se fijó en 5 mm, como requerimiento para mantener proporciones compatibles con la resolución del sistema de visión del Robotat.
- Se estableció que cada base de carga debía contar con un patrón único, de tal manera que el Robotat pudiera asignarle un ID exclusivo a cada cargador Tozo, como en la Figura 38.

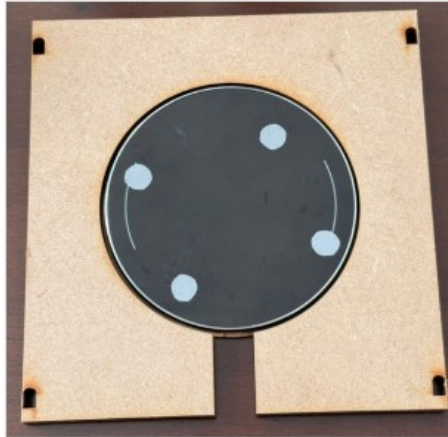
Figura 38. Patrones para estación de carga



Nota. Patrones que se utilizarán para que los drones CrazyFlie 2.1 detecte junto con el ecosistema Robotat. Elaboración propia.

El diseño del patrón debía ajustarse a la geometría superior del cargador Qi Tozo para permitir una calibración correcta dentro del ecosistema Robotat, como se evidencia en la identificación del marker 90 en la Figura 39.

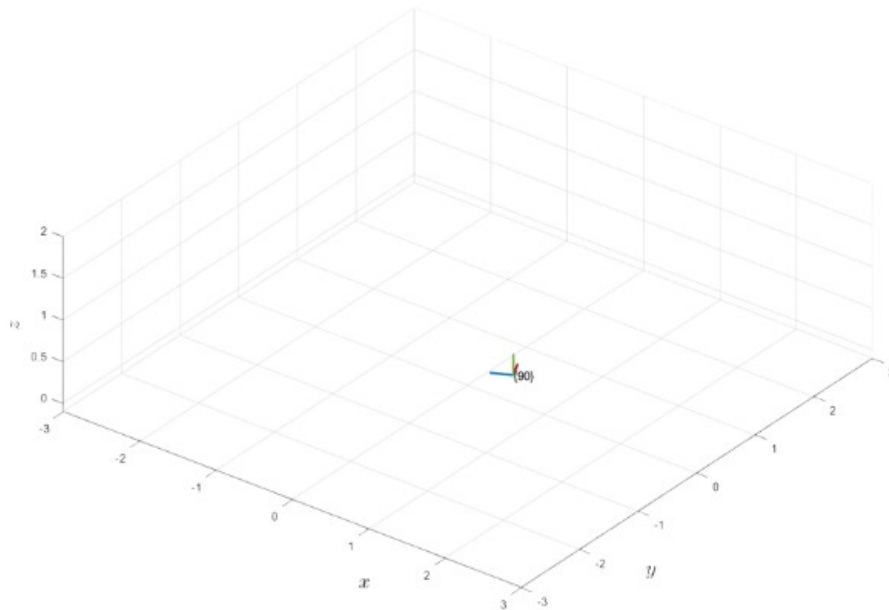
Figura 39. Patrón en estación de carga



Nota. Cargador inalámbrico Qi Tozo con círculos de cinta adhesiva y asignación del marker 90 en el Robotat. Elaboración propia.

En este caso, en la Figura 40 se muestra el marker 90 y que el ecosistema Robotat cumplió en poder detectarlo.

Figura 40. Marker 90 visualizado en Robotat



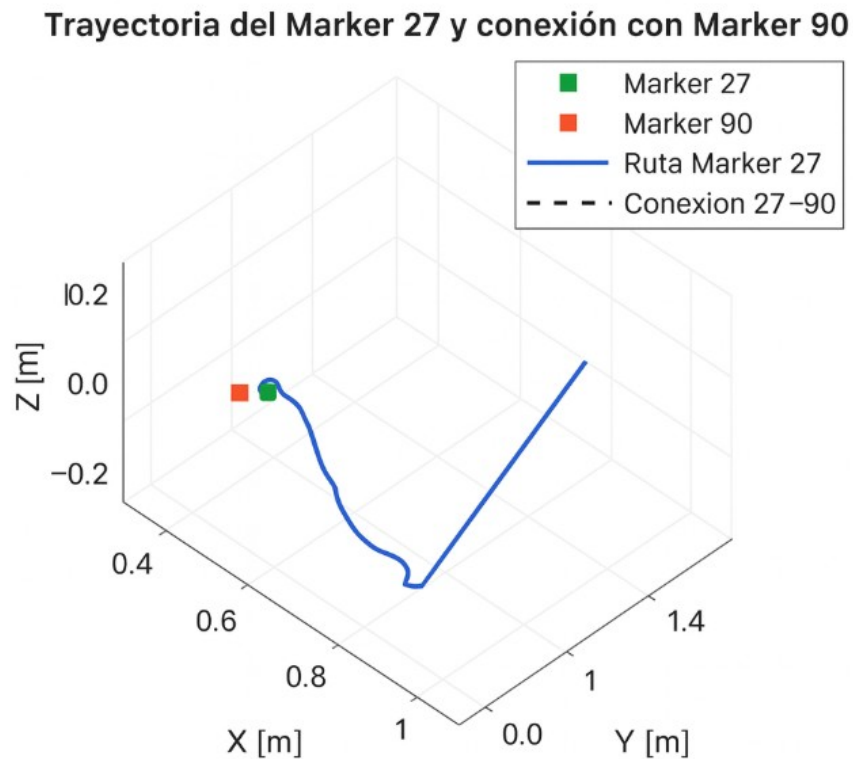
Nota. Detección del marker 90 por el ecosistema Robotat. Elaboración propia.

Otra forma de verificar que el Ecosistema Robotat detecta correctamente los markers correspondientes al dron (ID 27) y a la estación de carga (ID 90) es mediante una

visualización en tiempo real, como se muestra en el siguiente fragmento de código y en la Figura 41 , donde se grafican simultáneamente ambas posiciones:

```
1 robotat = robotat_connect();
2 if isempty(robotat)
3 error(' No se pudo conectar al servidor Robotat. ');
4 else
5 disp(' Conectado correctamente al servidor Robotat. ');
6 end
7
8 id_inicio = 27;
9 id_destino = 90;
```

Figura 41. Trayectoria del marker 27 y conexión con marker 90



Nota. Visualización del marker 27 y marker 90 detectados correctamente por el Ecosistema Robotat. Elaboración propia.

Visualización del dron Crazyflie (marker 27) y la estación de carga (marker 90) detectados correctamente por el sistema Robotat.

Asimismo, para obtener las coordenadas precisas de la estación de carga, se emplea el siguiente código, que solicita continuamente la posición del marker a Robotat y la

imprime en consola:

```
1
2     addpath('robotat');
3     r = robotat_connect();
4     marker_id = 90;
5
6     while true
7         pos = r.getMarker(marker_id);
8         if isempty(pos)
9             fprintf('Marker %d no detectado\n', marker_id);
10        else
11            fprintf('Marker %d -> x: %.3f, y: %.3f, z: %.3f\n', ...
12                marker_id, pos(1), pos(2), pos(3));
13        end
14        pause(0.05);
15    end
```

Las coordenadas obtenidas con este procedimiento se emplean posteriormente en el sistema de comunicación MQTT, permitiendo que el dron identifique la ubicación de la estación de carga para realizar el vuelo de llegada autónoma.

Durante las pruebas, se tomaron como referencia los experimentos desarrollados por Pablo Caal [9], en los cuales permite un vuelo controlado que integraba la comunicación entre el sistema de localización Robotat. Estos experimentos permitieron validar las condiciones necesarias para la detección confiable del dron y la estación de carga.

Se implementó un script de vuelo controlado que integra la comunicación entre el sistema de localización Robotat y el dron Crazyflie 2.1 equipado con el módulo Flow Deck. El Robotat proporcionaba en tiempo real las coordenadas espaciales del marker adherido al dron (ID 27), las cuales son utilizadas por el programa en MATLAB para monitorear y validar cada etapa del vuelo: detección inicial, despegue, desplazamiento corto y aterrizaje suave. Aunque se intentó prescindir del Flow Deck para que el control dependiera únicamente de las lecturas externas de Robotat, esto no fue posible de manera directa. Para eliminar la dependencia del Flow Deck y permitir el control del dron únicamente con la información de posición enviada por Robotat, es necesario implementar el sistema de comunicación MQTT, el cual se explicará en el siguiente capítulo.

Optimización de herramientas de software para la comunicación y control del dron CrazyFlie 2.1

Este capítulo describe la integración entre MATLAB, el ecosistema Robotat y el dron CrazyFlie 2.1 para habilitar la comunicación en tiempo real necesaria para el vuelo autónomo hacia la estación de carga diseñada en el capítulo anterior. Se presentan los métodos de conexión, el procesamiento de datos del sistema de captura de movimiento, el envío de posicionamiento externo al dron mediante MQTT y la evaluación del rendimiento del sistema durante pruebas experimentales.

9.1. Integración de Matlab con el ecosistema Robotat

En esta sección se explicará cómo se conecta MATLAB con Robotat y la lógica de comunicación. Para que todo funcione bien, se debe tener los archivos donde están las funciones correspondientes del Robotat y el dron CrazyFlie 2.1 para que funcione correctamente.

Para establecer la conexión con el Robotat, se empleó la función `robotat_connect()`, como se muestra en el siguiente fragmento de código.

```
1 addpath('robotat');  
2 robotat = robotat_connect();
```

Para conectarse al dron CrazyFlie 2.1 se debe colocar lo siguiente:

```
1 addpath('crazyflie');  
2 dron_id = 8;  
3 cf = crazyflie_connect(dron_id);
```

9.2. Procesamiento y transmisión eficiente de datos al dron CrazyFlie 2.1

La conexión inicial con el Robotat puede realizarse en MATLAB; sin embargo, la implementación del procesamiento MQTT y del control autónomo se desarrolló en Python debido que la API del CrazyFlie ofrece una integración más directa con el estimador de posición.

El código base utilizado para esta sección fue desarrollado originalmente por César Adrian Schwendener Morales con su trabajo titulado "Desarrollo de herramientas de software para la evaluación de sensores de posicionamiento en el control individual y seguro del cuadricóptero Crazyflie 2.1", cuya tesis se enfocó en la transmisión de datos del sistema de captura de movimiento hacia el dron CrazyFlie 2.1 mediante el protocolo MQTT. Este programa fue compartido como referencia y punto de partida para la continuación del trabajo experimental.

A partir del código inicial, se realizaron modificaciones con el objetivo de adaptar el sistema a las pruebas requeridas en este proyecto, específicamente para que el dron ejecutara un vuelo autónomo siguiendo las coordenadas del **marker 90**, el cual representa la posición de la estación de carga dentro del entorno del Ecosistema Robotat.

Se definen las variables globales que almacenan la posición del dron, su trayectoria, la conexión con el CrazyFlie y las coordenadas del marker 90.

```
1 # -----
2 # VARIABLES GLOBALES
3 # -----
4 mocap_pose = {'x': 0.0, 'y': 0.0, 'z': 0.0}
5 real_trajectory = []
6 cf = None
7 last_ts = None
8
9 marker90 = {'x': 0.131, 'y': -1.032, 'z': 0.012}
```

La transmisión de datos se realiza de forma continua a través de un hilo que escucha el tópico mocap/drone2 del servidor MQTT, donde se publican las posiciones estimadas por el MoCap. Cada mensaje recibido se procesa en la función `on_message()`, que decodifica el formato JSON, filtra los datos antiguos y actualiza la posición del dron en tiempo real:

```
1 # -----
2 # CALLBACK MQTT
3 # -----
4 def on_message(client, userdata, msg):
5     global mocap_pose, cf, last_ts
6     try:
7         data = json.loads(msg.payload.decode())
8         pos = data['payload']['pose']['position']
9         ts_str = data.get('ts', None)
10
11     if ts_str is not None:
12         msg_time = datetime.fromisoformat(ts_str.replace('Z', '+00:00'))
13         if last_ts is not None and msg_time <= last_ts:
```

```

14 return
15 last_ts = msg_time
16
17 mocap_pose['x'] = float(pos['x'])
18 mocap_pose['y'] = float(pos['y'])
19 mocap_pose['z'] = float(pos['z'])

```

De esta forma, cada posición medida por el sistema MoCap se transmite inmediatamente al estimador Kalman interno de CrazyFlie mediante el comando `cf.extpos.send_extpos()`, garantizando una actualización rápida y precisa de la posición del dron.

Esta estructura permite una comunicación eficiente entre el sistema de percepción y el controlador de vuelo, reduciendo la latencia y mejorando la respuesta del sistema durante el vuelo autónomo hacia el marker 90, correspondiente a la estación de carga.

9.3. Pruebas de llegada autónoma a la estación de carga

Para esta prueba de vuelo, se modificó el código base incluyendo la función `fly_to_marker()`, que define la secuencia completa de despegue, vuelo y aterrizaje sobre el punto objetivo:

```

1 def fly_to_marker(cf):
2     commander = HighLevelCommander(cf)
3     hover_height = 0.5
4     target_x, target_y, target_z = marker90['x'],
5     marker90['y'], hover_height
6
7     commander.takeoff(hover_height, 3.0)
8     time.sleep(3.5)
9
10    commander.go_to(target_x, target_y, target_z, 0.0,
11    0.6, relative=False)
12    time.sleep(4.0)
13
14    commander.land(0.0, 2.0)
15    time.sleep(3.0)
16    commander.stop()

```

Durante las pruebas, el dron realiza correctamente la secuencia de despegue y desplazamiento hacia el marker 90, manteniendo una altura constante de 0.5 m. Al llegar al destino se efectúa un vuelo estacionario breve antes de iniciar el aterrizaje controlado.

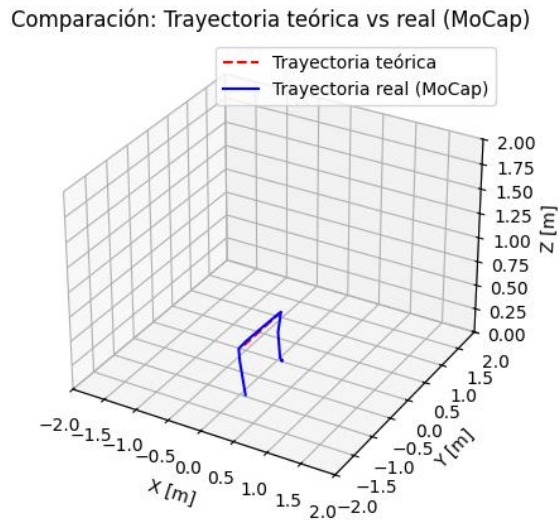
9.4. Evaluación del rendimiento y tiempo de respuesta del sistema

Para evaluar el rendimiento del sistema y el tiempo de respuesta de la transmisión de datos, se analizaron las trayectorias registradas durante el vuelo autónomo hacia el marker 90, como se muestra en la Figura 42.

Cada posición estimada por el sistema MoCap fue almacenada en la lista `real_trajectory` y posteriormente graficada junto a la ubicación final esperada, permitiendo visualizar la precisión del recorrido:

```
1 real_trajectory.append([
2 mocap_pose['x'],
3 mocap_pose['y'],
4 mocap_pose['z']
5 ])
```

Figura 42. Trayectoria del dron CrazyFlie 2.1 con el sistema MQTT

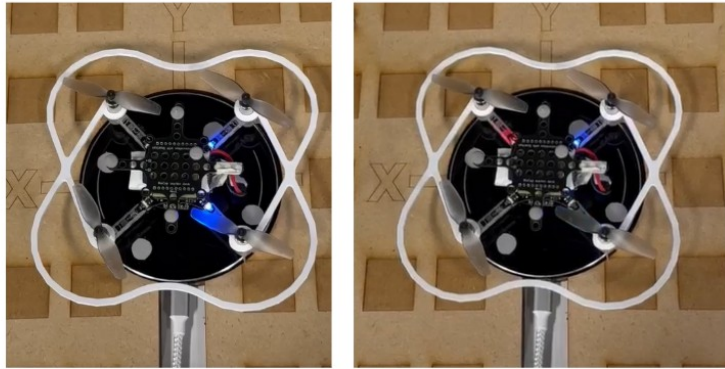


Nota. El análisis de la trayectoria muestra una correspondencia cercana entre el movimiento real del dron y la posición objetivo, indicando una correcta sincronización entre el sistema de captura de movimiento y el estimador del CrazyFlie. Elaboración propia.

9.5. Comprobación de la carga en la estación de carga

Para verificar que el dron CrazyFlie 2.1 se está cargando correctamente, uno de los LEDs azules del dron debe parpadear. Si el dron está apagado, no funcionará, por lo tanto, debe mantenerse encendido durante el proceso y posicionado en centro del cargador inalámbrico, como se muestra en la Figura 43.

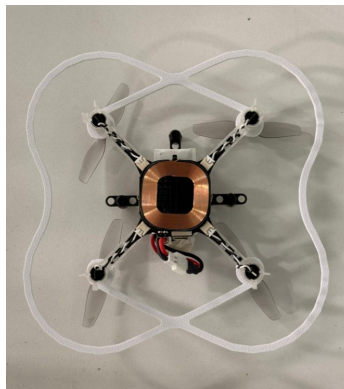
Figura 43. Comprobación visual de carga en la estación del Crazyflie 2.1



Nota. Durante la carga, uno de los LED azules del dron parpadea, confirmando la conexión correcta con la estación. Elaboración propia.

Para que el proceso de carga funcione, es necesario retirar el Flow Deck y reemplazarlo por el Qi 1.2 charger deck, como se muestra en la Figura 44.

Figura 44. Instalación del Qi 1.2 charger deck en el dron Crazyflie 2.1



Nota. El charger deck permite la carga inalámbrica del dron y se conecta en el mismo puerto donde usualmente se coloca el Flow Deck. Elaboración propia.

Los resultados demostraron que la integración entre el Robotat MQTT y el CrazyFlie 2.1 permitió un control autónomo estable y una detección confiable de la estación de carga. La trayectoria medida se mantuvo próxima al objetivo y la latencia del sistema fue suficientemente baja para permitir el aterrizaje controlado sobre el cargador Qi Tozo, validando la eficacia del sistema implementado.

- Las guardas protectoras redujeron significativamente los daños por colisiones, pasando de un promedio de cuatro hélices rotas por semana a cero tras su implementación. Este resultado valida la viabilidad del uso de manufactura digital como solución efectiva para extender la vida útil del CrazyFlie 2.1 bajo condiciones de uso intensivo.
- El desarrollo de guardas de protección mediante manufactura digital rápida contribuyó a reducir riesgos durante la experimentación, incrementando la seguridad operativa del dron y del entorno de trabajo.
- El diseño final de las guardas demostró ser estructuralmente adecuado, soportando caídas de hasta ocho metros sin deformaciones permanentes, lo cual preserva la capacidad de maniobra del dron. Esto confirma que los criterios geométricos definidos son compatibles con los límites de carga y estabilidad del CrazyFlie 2.1.
- Las estaciones de carga inalámbrica Qi mostraron una eficiencia consistente, logrando una alineación exitosa entre el dron y el cargador para varios aterrizajes automáticos. Estos resultados indican que el sistema es viable para ciclos repetitivos de operación, siempre que las tolerancias de posicionamiento se mantengan dentro del rango detectado por el Robotat.
- La integración entre el Robotat, MQTT y el CrazyFlie 2.1 permitió un control autónomo seguro. Esto respalda la factibilidad del vuelo autónomo hacia estaciones de carga dentro del ecosistema Robotat, bajo las condiciones de conectividad empleadas.
- Python resultó más eficiente que Matlab para el control autónomo, ya que permitió enviar estimaciones externas a una frecuencia superior, haciendo viable el control reactivo del dron basado únicamente en datos del Robotat.

- Se sugiere explorar nuevos materiales y técnicas de manufactura que permitan reducir el peso de las guardas de protección sin comprometer su resistencia. Eso permitiría mantener la seguridad alcanzada, disminuyendo el impacto en el consumo energético del dron.
- Se plantea modificar el diseño de las guardas de manera que incluyan una cobertura superior parcial, con el fin de proteger las hélices y la estructura del dron en la parte superior. Esto reduciría la probabilidad de daños en colisiones con objetos por encima del dron, manteniendo la ligereza de la estructura.
- Se aconseja sustituir o reforzar el material de la caja de la estación de carga inalámbrica con el fin de incrementar su durabilidad y resistencia a impactos o manipulación constante en el laboratorio.
- Se considera pertinente continuar con pruebas experimentales más prolongadas para evaluar el comportamiento tanto de las guardas como de la estación de carga bajo condiciones de uso repetitivo. Esto permitirá identificar posibles puntos de falla y realizar mejoras de diseño en futuras iteraciones.
- Se propone diseñar una interfaz gráfica que permita visualizar en tiempo real el estado operativo del dron, incluyendo parámetros como nivel de conexión y posición dentro del Robotat. Esta herramienta facilitaría la supervisión del sistema durante las pruebas, al verificar que la estación Qi funcione correctamente y permitir detectar fallas o desconexiones de manera inmediata. Además, dicha interfaz podría integrarse con los datos transmitidos por MQTT para un control más intuitivo y completo.
- Se sugiere desarrollar rutinas que le permitan al dron ejecutar ciclos automáticos de despegue, vuelo, aterrizaje y carga sin intervención humana. Esto permitiría validar algoritmos de control autónomo y de planificación de trayectorias en tiempo real.

- [1] B. M. Parra Toro, *Diseño y validación de algoritmos de control para el dron Crazyflie 2.0*, https://www.uned.es/universidad/facultades/dam/jcr:6a44c5da-a948-4a69-bf61-a501739216b6/PFM_BlancaM_Parra_Toro.pdf, Universidad Nacional de Educación a Distancia, 2021.
- [2] F. Sanabria, *Diseño e implementación de una plataforma de pruebas para sistemas de control para el dron Crazyflie 2.0*, <https://repositorio.uvg.edu.gt/handle/123456789/4510>, Tesis de licenciatura, 2022.
- [3] C. Perafán, *Robotat: un ecosistema robótico de captura de movimiento y comunicación inalámbrica*. <https://repositorio.uvg.edu.gt/handle/123456789/4250>, Tesis de licenciatura, 2022.
- [4] J. Gordillo, *Diseño e implementación de un paquete de herramientas de software para controlar inalámbricamente un enjambre de drones Crazyflie dentro de un ecosistema robótico basado en captura de movimiento*, <https://repositorio.uvg.edu.gt/handle/123456789/4906>, Tesis de licenciatura, 2023.
- [5] D. Oztzy, *Uso del cuadricóptero Crazyflie dentro del ecosistema Robotat para la generación de trayectorias y la evasión de obstáculos*, <https://repositorio.uvg.edu.gt/handle/123456789/4835>, Tesis de licenciatura, 2023.
- [6] C. Developers, *CrazySwarm Documentation*, <https://crazyswarm.readthedocs.io/en/latest/>, n.d.
- [7] B. Garrido, *Levantamiento de una plataforma de pruebas para sistemas multidrones con OptiTrack y CrazySwarm*, <https://repositorio.uvg.edu.gt/handle/123456789/5130>, Tesis de licenciatura, 2023.
- [8] J. Avila, *Adaptación del sistema de drones CrazySwarm al ecosistema Robotat*, <https://repositorio.uvg.edu.gt/xmlui/handle/123456789/5115>, Tesis de licenciatura, 2023.

- [9] P. Caal, *Desarrollo de herramientas de software para el control individual y seguro del cuadricóptero Crazyflie 2.1 utilizando la placa de expansión de posicionamiento con odometría visual Flow Deck*, <https://teams.microsoft.com/v2/>, Tesis de licenciatura, 2024.
- [10] F. J. Mañas-Álvarez, M. Guinaldo, R. Dormido, R. Socas y S. Dormido, *Control basado en eventos mediante umbral relativo aplicado al control de altitud de cuadricópteros Crazyflie 2.1*, <https://ruc.udc.es/dspace/handle/2183/28331>, Tesis de máster, 2021.
- [11] Bitcraze, *Crazyflie 2.1*, <https://www.bitcraze.io/products/old-products/crazyflie-2-1/>, 2021.
- [12] B. AB, *The Coordinate System of the Crazyflie 2.x*, <https://www.bitcraze.io/documentation/system/platform/cf2-coordinate-system/>, 2024.
- [13] Bitcraze AB, *Crazyradio PA*, <https://www.bitcraze.io/products/crazyradio-pa/>, 2024.
- [14] Bitcraze AB, *Flow Deck V2*, <https://www.bitcraze.io/products/flow-deck-v2/>, 2024.
- [15] B. AB, *Python API explanation*, <https://www.bitcraze.io/documentation/>, 2025.
- [16] Bitcraze AB, *Qi 1.2 Charger Deck*, https://www.bitcraze.io/products/qi-1_2-charger-deck/, 2024.
- [17] Amazon, *Cargador inalámbrico actualizado - Almohadilla compatible con Qi*, <https://www.amazon.com/dp/B0D4HCT8FR>, 2024.
- [18] Bitcraze, *Motion capture marker deck*, <https://www.bitcraze.io/products/motion-capture-marker-deck/>, 2024.
- [19] OptiTrack, *Markers*, <https://docs.optitrack.com/motive/markers>, 2024.
- [20] B. AB, *Motion Capture positioning*, <https://www.bitcraze.io/documentation/system/positioning/mocap-positioning/>, 2025.
- [21] B. AB, *4 x 7 mm DC-motor pack for Crazyflie 2.X*, <https://store.bitcraze.io/products/4-x-7-mm-dc-motor-pack-for-crazyflie-2>, 2023.
- [22] B. AB, *4 x 7x20 mm DC-motor pack for Crazyflie 2.X (thrust upgrade)*, <https://store.bitcraze.io/products/4-x-7x20-mm-dc-motor-pack-for-crazyflie-2-x-thrust-upgrade>, 2023.
- [23] B. AB, *Crazyflie 2.X: 4 x spare 7 mm motor mounts*, <https://store.bitcraze.io/products/crazyflie-2-0-4-x-spare-7-mm-motor-mounts>, 2023.
- [24] B. AB, *Propeller 47-17 (4CCW+4CW) (black)*, <https://store.bitcraze.io/products/propeller-47-17-black>, 2023.
- [25] HQProp, *HQ Ultralight Prop 2X0.9 (51MM) Grey(2CW+2CCW)-PC-1.5MM Shaft*, <https://www.hqprop.com/hq-ultralight-prop-2x09-51mm-grey2cw2ccw-pc-15mm-shaft-p0457.html>, 2023.
- [26] B. AB, *Datasheet Crazyflie 2.1 - Rev 3*, https://www.bitcraze.io/documentation/hardware/crazyflie_2_1/crazyflie_2_1-datasheet.pdf, 2021.
- [27] buildsrobots, *Crazyflie Propeller Guard*, <https://www.thingiverse.com/thing:1151252>, Publicado en Thingiverse., 2015.

Los anexos proporcionan el material técnico complementario del proyecto, incluyendo códigos fuente, modelos 3D y archivos de manufactura digital necesarios para reproducir los experimentos. Se incluyen enlaces a repositorios GitHub con los scripts en Matlab/Python, los diseños de las guardas y la estación de carga, así como los archivos finales utilizados para la fabricación de los prototipos.

13.1. Código fuente

13.1.1. Código de Matlab/Python

El código desarrollado para la experimentación se encuentra disponible en el siguiente repositorio de GitHub. Este enlace contiene los scripts en Matlab/Python, así como la documentación necesaria para su ejecución dentro del ecosistema Robotat.

<https://github.com/angie0202/Trabajo-de-Graduacion-Angela-Figueroa-en-MATLAB>

13.1.2. Diseño y planos

En el siguiente enlace de GitHub se encuentran disponibles los archivos en formato STL, correspondientes a los diseños de las guardas y la estación de carga:

<https://github.com/angie0202/Trabajo-de-Graduacion-Angela-Figueroa-en-Inventor.git>

Para acceder al archivo listo para impresión 3D o corte láser, se puede utilizar el siguiente enlace de GitHub:

<https://github.com/angie0202/Trabajo-de-Graduacion-Angela-FigInventor-Manufactura>